}

TERRAFORM

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}

# Configure the AWS Provider
provider "aws" {
  region = "us-east-1"
}

# Create a VPC
resource "aws_vpc" "example" {
  cidr_block           = "10.0.0.0/16"
  enable_dns_support   = true
  enable_dns_hostnames = true

  tags = {
    Name = "ExampleVPC"
  }
}
```

```
# Create Subnet 1 (Public)
resource "aws_subnet" "subnet1" {
  vpc_id                  = aws_vpc.example.id
  cidr_block              = "10.0.1.0/24"
  map_public_ip_on_launch = true
  availability_zone       = "us-east-1a"

  tags = {
    Name = "Subnet1-Public"
  }
}

# Create Subnet 2 (Private)
resource "aws_subnet" "subnet2" {
  vpc_id            = aws_vpc.example.id
  cidr_block        = "10.0.2.0/24"
  availability_zone = "us-east-1b"

  tags = {
    Name = "Subnet2-Private"
  }
}

# Create an Additional Public Subnet
resource "aws_subnet" "public" {
  vpc_id                  = aws_vpc.example.id
  cidr_block              = "10.0.3.0/24"
  map_public_ip_on_launch = true
  availability_zone       = "us-east-1c"

  tags = {
    Name = "PublicSubnet"
  }
}

# Create an Internet Gateway
resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.example.id

  tags = {
    Name = "InternetGateway"
  }
}

# Create a Route Table for Public Subnets
resource "aws_route_table" "public_rt" {
  vpc_id = aws_vpc.example.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.igw.id
  }

  tags = {
    Name = "PublicRouteTable"
  }
}

# Associate Route Table with Public Subnet 1
resource "aws_route_table_association" "subnet1_association" {
  subnet_id      = aws_subnet.subnet1.id
  route_table_id = aws_route_table.public_rt.id
}

# Associate Route Table with Public Subnet (Additional)
resource "aws_route_table_association" "public_association" {
  subnet_id      = aws_subnet.public.id
  route_table_id = aws_route_table.public_rt.id
}

# Create a Security Group for SSH Access
resource "aws_security_group" "allow_ssh" {
  vpc_id = aws_vpc.example.id

  ingress {
    description = "Allow SSH"
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "AllowSSH"
  }
}

# Create an EC2 Instance in Subnet 1 (Public)
resource "aws_instance" "example1" {
  ami                    = "ami-0c55b159cbfafe1f0"  # Change this to your preferred AMI
  instance_type          = "t2.micro"
  subnet_id              = aws_subnet.subnet1.id
  security_groups        = [aws_security_group.allow_ssh.name]
  associate_public_ip_address = true

  tags = {
    Name = "ExampleInstance1"
  }
}

# Create an EC2 Instance in Subnet 2 (Private)
resource "aws_instance" "example2" {
  ami                 = "ami-0c55b159cbfafe1f0"  # Change this to your preferred AMI
  instance_type        = "t2.micro"
  subnet_id            = aws_subnet.subnet2.id
  security_groups       = [aws_security_group.allow_ssh.name]

  tags = {
    Name = "ExampleInstance2"
  }
}

# Create an EC2 Instance in the Public Subnet
resource "aws_instance" "example3" {
  ami                 = "ami-0c55b159cbfafe1f0"  # Change this to your preferred AMI
  instance_type        = "t2.micro"
  subnet_id            = aws_subnet.public.id
  security_groups       = [aws_security_group.allow_ssh.name]
  associate_public_ip_address = true

  tags = {
    Name = "ExampleInstance3"
  }
}
#terraform init
#terraform validate
#terraform plan
#terraform apply
#terraform destroy
```

Cheatsheet - https://registry.terraform.io/providers/hashicorp/aws/latest/docs